

# Release Notes for Platform LSF®

## Version 6.1

April 8 2005

Platform Computing

Comments to: [doc@platform.com](mailto:doc@platform.com)

Contents	◆	“What’s New in the Platform LSF Version 6.1” on page 2
	◆	“Upgrade and Compatibility Notes” on page 11
	◆	“Known Issues” on page 22
	◆	“Learning About Platform LSF Version 6.1” on page 23
	◆	“Getting Technical Support” on page 26

## What's New in the Platform LSF Version 6.1

- ◆ “Platform LSF”
  - ❖ “High performing, open scalable architecture”
  - ❖ “Comprehensive intelligent scheduling policies”
  - ❖ “Reliability and usability”
- ◆ “Platform LSF License Scheduler”
  - ❖ “Failover provisioning in LANs and WANs”
  - ❖ “License ownership and distribution”
  - ❖ “Administration and management”
- ◆ “Platform LSF Reports”
  - ❖ “Administration and management”
  - ❖ “Data collection”
- ◆ “Platform LSF Analytics”
  - ❖ “Administration and management”
  - ❖ “Data collection”
  - ❖ “Database integration”
  - ❖ “Out-of-box cubes”
- ◆ “Platform LSF Desktop Support”
  - ❖ “New operating system support”
  - ❖ “Security enhancements”
  - ❖ “Usability”

## Platform LSF

### High performing, open scalable architecture

- ◆ The primary focus of Platform LSF Version 6.1 is to enhance performance and scalability:
  - ❖ Faster user interactions for job submission and query
  - ❖ Accelerated time to schedule and dispatch jobs
  - ❖ Quicker reconfiguration of master failover for higher availability
- ◆ With an optimally tuned and configured environment, you can benefit from the following enhancements:
  - ❖ Up to 5,000 hosts per cluster
  - ❖ 500,000 active jobs per cluster at any one time
  - ❖ More concurrent users continuously executing Platform LSF commands
  - ❖ 1 million completed jobs per day
  - ❖ 40,000 active application licenses can be managed
- ◆ Dynamic addition of hosts to the cluster has been improved to include dynamic host group membership. This shortens the time to add and remove hosts. The feature is disabled by default.

To enable dynamic host configuration, you must define the following parameters:

  - ❖ LSF\_MASTER\_LIST and LSF\_DYNAMIC\_HOST\_WAIT\_TIME in `lsf.conf`

❖ LSF\_HOST\_ADDR\_RANGE in `lsf.cluster.cluster_name`

LSF\_DYNAMIC\_HOST\_TIMEOUT in `lsf.conf` is obsolete.

Use the `badmin` command to add dynamic hosts to the host group or remove dynamic hosts from the host group (You need to run `mbd reconfig` for these changes to take effect).

The following example shows the command to add the dynamic host `host_name` to the host group `host_group`:

```
badmin hghostadd [-C "message"] [-f] host_group
"host_names"
```

The following example shows the command to remove the dynamic host group `host_name` from the host group `host_group`:

```
badmin hghostdel [-C "message"] [-f] host_group
"host_names"
```

At startup, slave hosts wait to receive the acknowledgement from the master LIM. This acknowledgement indicates to the slave host that it has already been added to the cluster. If it does not receive the acknowledgement within the time specified by LSF\_DYNAMIC\_HOST\_WAIT\_TIME in `lsf.conf`, the slave host contacts the master LIM to add itself to the cluster.

## Comprehensive intelligent scheduling policies

- ◆ Where a group of hosts has a critical resource, you can assign that resource as an *exclusive resource* and allow only jobs that require that resource to run on those hosts. To maximize availability of hosts to meet specific application requirements and guarantee that the most important jobs get the resources that they need, the specified host will not be selected unless the job explicitly requests the exclusive resource in its resource requirements. Jobs not requiring the exclusive resource will never be dispatched to those hosts.

An exclusive resource is defined in `lsf.shared` and assigned to a host in `lsf.cluster.cluster_name`. For example the resource, `bigmem` is assigned to the `hostE` host:

```
Begin Host
HOSTNAME model type server rlm mem pg RESOURCES
hostE    !      !      1      ()  ()  () (!bigmem)
...
End Host
```

A job must explicitly request `bigmem` in either the job or queue level resource requirement to be dispatched to `hostE`. The following commands request the `bigmem` resource, and will be dispatched to `hostE`:

- ❖ `bsub -R "bigmem" myjob`
- ❖ `lsplace -R "bigmem" myjob`
- ❖ `lsgrun -R "bigmem" myjob`
- ❖ `lsrun -R "bigmem" myjob`

Exclusive resources are only applicable in placement or scheduling scenarios. Commands such as `lsload`, `lshosts`, or `bhosts` still list the host even if the exclusive resource is not explicitly specified with the `-R` option.

- ◆ Slot-based parallel job scheduling—by default, you cannot specify more slots than the system's eligible number of processors. Since you can define more than one slot for each CPU, slot-based job scheduling allows you to submit parallel jobs based on the number of available slots rather than processors.

For example, `hostA` has 2 processors, but has 4 slots defined:

```
% lshosts
HOST_NAME  type  model    cpuf    ncpus  maxmem  maxswp  server  RESOURCES
hostA      SOL64  Ultra5F  18.6    2      128M    594M    Yes    ()
```

However, you cannot submit a parallel job that requests more than 2 processors

```
% bsub -n 4 myjob
```

Too many processors requested. Job not submitted.

But you can submit two jobs requesting two slots each and both of them can run at the same time.

When `PARALLEL_SCHED_BY_SLOT=Y` is defined in `lsb.params`, parallel jobs can request the number of slots they require.

- ◆ Resource reservation “OR” support in the `rusage` string—use the OR operator (`|`) in resource requirement strings to specify multiple resource reservations and evaluate reservations in the specified order until one is satisfied. This is useful when a job can run with more than one `rusage` string and you want LSF to select the most appropriate resource reservation to meet the requirements for the job.

For example, you are running an application version 1.5 as a resource called `app_lic_v15` and the same application version 2.0.1 as a resource called `app_lic_v201`. The license key for version 2.0.1 is backward compatible with version 1.5, but the license key for version 1.5 will not work with 2.0.1.

Job-level resource requirement specifications that use the `|` operator take precedence over any queue-level resource requirement specifications.

If you can only run your job using one version of the application, submit the job without specifying an alternate resource. To submit a job that will only use `app_lic_v201`:

```
% bsub -R "rusage[app_lic_v201=1]" myjob
```

If you can run your job using either version of the application, try to reserve version 2.0.1 of the application. If it is not available, you can use version 1.5. To submit a job that will try `app_lic_v201` before trying `app_lic_v15`:

```
% bsub -R "rusage[app_lic_v201=1|app_lic_v15=1]" myjob
```

If different versions of an application require different system resources, you can specify other resources in your `rusage` strings. To submit a job that will use 20 MB of memory for `app_lic_v15` or 20 MB of memory and 50 MB of swap space for `app_lic_v201`:

```
% bsub -R "rusage[mem=20:app_lic_v15=1|mem=20:swp=50:app_lic_v201=1]" myjob
```

**Reliability and usability**

- ◆ Pending job management—limits the number of pending jobs by setting cluster wide, user level and user group level settings. This prevents the overloading of the cluster.
- ◆ Use the `-oo` and `-oe` options of `bsub` and `bmod` to overwrite the LSF output and error files if they already exist. The original behavior of appending to the output and error files is still available with the `-o` and `-e` options.
- ◆ Administrator control over whether users can use `btop` and `bbot` to move jobs to the top and bottom of queues. When `JOB_POSITION_CONTROL_BY_ADMIN = Y`, only LSF administrators (including any queue administrators) can use `bbot` and `btop` to move jobs within a queue. A user attempting to use `bbot` or `btop` receives the error message `User permission denied`.
- ◆ Using host name ranges as aliases—the default host file syntax  
`ip_address official_name [alias [alias ...]]`  
 is powerful and flexible, but it is difficult to configure in systems where a single host name has many aliases, and in multihomed host environments.  
 In these cases, the `hosts` file can become very large and unmanageable, and configuration is prone to error.  
 The syntax of the LSF `hosts` file supports host name ranges as aliases for an IP address. This simplifies the host name alias specification.  
 To use host name ranges as aliases, the host names must consist of a fixed node group name prefix and node indices, specified in a form like:  
`host_name[index_x-index_y, index_m, index_a-index_b]`  
 For example:  
`atlasD0[0-3,4,5-6, ...]`  
 is equivalent to:  
`atlasD0[0-6, ...]`

## Platform LSF License Scheduler

**Failover provisioning in LANs and WANs**

You can configure LSF License Scheduler for enhanced performance, easy organization, and reliable license distribution.

You need only one host to run the LSF License Scheduler but you can configure your site for a failover mechanism where multiple candidate hosts can take over the scheduling in case of a failure. This configuration can be used in a local network (LAN) or across multiple sites in a wide-area network (WAN).

License scheduling across sites can be streamlined because LSF License Scheduler supports service provisioning during breaks in wide area network connections. This allows you to run LSF License Scheduler from one host that controls license scheduling across multiple sites.

**License ownership and distribution**

- ◆ Slot release on license preemption—slots will no longer be held for a job when a license has been preempted.  
 This is enabled by default, but can be disabled by setting the following keyword in the `lsf.conf` file:

```
LSF_LIC_SCHED_PREEMPT_SLOT_RELEASE = n
```

- ◆ Default License Scheduler projects—allows job submission without an explicit project assigned.

If a user submits a job without specifying a license project, it will be assigned to the default project. This default project does not need to be explicitly defined within the Begin/End Projects section in the `lsf.licensescheduler` file.

Use the reserved keyword `default` to configure the default project within a distribution policy. The following example shows how to use `default` to mean all other jobs not specified to be in `proj1` or `proj2`:

```
Begin Feature
NAME = vcs
DISTRIBUTION = LanServer(proj1 3 proj2 1 default 1)
End Feature
```

You can also use this default license project to configure features that do not have a scheduling policy, and will simply track the license usage of each job in a “read-only” mode. The following example shows a feature which will allocate all the available license tokens for a job submitted without a license project:

```
Begin Feature
NAME = vcs
DISTRIBUTION = LanServer(default 1)
End Feature
```

- ◆ Private licenses—a particular project owns the license and has exclusive rights to it. No other job can use this specified license.

Use the new optional keyword, `NON_SHARED_DISTRIBUTION` in the Feature section, and define the number for `private`. For example:

```
Begin Feature
NAME = f1 # total 15 on LanServer and 15 on WanServer
FLEX_NAME = VCSRruntime_Net
DISTRIBUTION = LanServer(p1 1 p2 1) WanServer(p1 1 p2 1/3)
NON_SHARED_DISTRIBUTION = LanServer(p1 10) WanServer (p1 5
p2 3)
PREEMPT_RESERVE=Y
End Feature
```

The new command option `blinfo -a` displays the configured shared licenses distribution and non-shared licenses distribution as well.

- ◆ New `-Lp` option for License Scheduler projects. This option is used for License Scheduler projects, while the current `-P` option is now used exclusively for LSF projects. This option is useful for distinguishing between LSF projects and License Scheduler projects. The `-Lp` option is available for the following commands:
  - ❖ `bsub -Lp lp_name` explicitly assigns projects to the License Scheduler
  - ❖ `bmod -Lp lp_name jobID` attaches the job to the license project
  - ❖ `bmod -Lpn lp_name jobID` detaches the job from the license project
  - ❖ `bjobs -Lp lp_name` displays all jobs belonging to the license project
  - ❖ `bhist -Lp lp_name` displays all jobs belonging to the license project
  - ❖ `bacct -Lp lp_name` displays all the jobs that ran in the license project
  - ❖ `blstat -Lp lp_name` displays all license usage in the license project

The -P option for this command is no longer available and is replaced by the -Lp option.

- ❖ `blinfo -Lp` displays all project names that LSF License Scheduler is managing

The -P option for this command is no longer available and is replaced by the -Lp option.

## Administration and management

- ◆ Multiple administrator support—more than one user is given privileges to perform administrative tasks.

Use the existing ADMIN keyword inside `lsf.licensescheduler` as a space-separated list of host names:

```
ADMIN = lsfadmin user1 user2 root
```

The first user specified in the list is the primary administrator, similar to an LSF primary administrator.

Only the primary administrator or root can run the License Scheduler daemon (`blsd`). By default, the primary administrator owns all the working files and directories created by License Scheduler.

The License Scheduler primary administrator should be the same as the LSF primary administrator. This is so the daemon will have permission to create the files and directories inside the LSF working directory.

If a non-privileged user tries to perform an administrative task, the License Scheduler daemon will send back a “permission denied” message to inform the user.

By default, the install script will peek at `lsf.cluster.cluster_name` and prompt you to use the LSF administrators for License Scheduler.

- ◆ Allocate shares of license features across clusters and between LSF jobs and non-LSF interactive jobs:
  - ❖ You can globally enable one share of all license features for interactive tasks. Set `ENABLE_INTERACTIVE=Y` in `lsf.licensescheduler`.  
By default, `ENABLE_INTERACTIVE` is not set. License Scheduler allocates licenses equally to each cluster and does not distribute licenses for interactive tasks.
  - ❖ You can configure the allocation of license shares to:
    - ❖ Change the share number between clusters for specific features
    - ❖ Limit the scope of license usage and change the share number between LSF jobs and interactive tasks for a feature

**To manage interactive (non-LSF) tasks in License Scheduler projects, you require the LSF Task Manager, `taskman`. The Task Manager utility is supported by but not shipped with License Scheduler. For more information about `taskman`, contact Platform.**

Set the `ALLOCATION` keyword in the Features section of `lsf.licensescheduler`. This feature ignores the global setting of the `ENABLE_INTERACTIVE` parameter because `ALLOCATION` is configured for the feature.



For example:

```
Begin Feature
NAME = ApplicationX
DISTRIBUTION = LicenseServer1 (project1 1)
ALLOCATION = project1 (Cluster1 1 Cluster2 1 interactive
1)
End Feature
```

## Platform LSF Reports

### Administration and management

- ◆ Administrative GUI—view, access, and perform all common administration and customization tasks in a single, consolidated GUI console.
- ◆ Administrative email notifications—email notification parameters can be specified and emails sent for all events, including when a cube or report is built.
- ◆ License feature vendor name mapping—license feature vendor names are displayed and mapped to the license cubes enabling reports to be configured.

### Data collection

- ◆ Data collection framework—new, lightweight method of collecting data from multiple clusters.
- ◆ Configuration data logging—collect data about LSF host group and user group configuration for data logging.
- ◆ Redesigned job statistics cubes—Platform LSF Reports will only collect data concerning batch jobs. Within these jobs are interactive and parallel jobs.

## Platform LSF Analytics

### Administration and management

- ◆ Administrative GUI—view, access, and perform all common administration and customization tasks in a single, consolidated GUI console.
- ◆ Administrative email notifications—email notification parameters can be specified and emails sent for all events, including when a cube or report is built.
- ◆ License feature vendor name mapping—license feature vendor names are displayed and mapped to the license cubes enabling reports to be configured.
- ◆ Automatic installation and deployment—two separate installations merged into a single installation. The Platform LSF environment variables are auto-detected during the installation of the console. Platform LSF Analytics automatically starts collecting data immediately.

### Data collection

- ◆ Data collection framework—new, lightweight method of collecting data from multiple clusters.
- ◆ Month/level/date dimension—“Month” dimension is now included in all cubes.
- ◆ Pending time ranking—Workload cube includes “seconds”-level ranking for pending time.
- ◆ UNIX exit code dimension—Workload cube collects data about the UNIX exit code.
- ◆ Configuration data logging—collect data about LSF host group and user group configuration for data logging.

- ◆ Redesigned job statistics cubes—Platform LSF Analytics will only collect data and statistics concerning batch jobs. Within these jobs are interactive and parallel jobs.
- Database integration**
- ◆ Daily reaggregation and data purge—Platform LSF Analytics has a daily table in the database. Every day, data is reaggregated into this table, and obsolete data is purged by default.

#### Out-of-box cubes

- ◆ Platform LSF Analytics has the following new license cubes:
  - ❖ License usage by user group
  - ❖ License utilization by feature
  - ❖ License denial by feature

## Platform LSF Desktop Support

### New operating system support

- ◆ Support for Windows 2000 with HP ProtectTools Authentication Services (WinSE)
- Security enhancements**
- ◆ Blacklist and whitelist—you can use a blacklist or a whitelist to control a desktop client's access to the desktop server to request work: to exclude only a few desktop clients from requesting work, use a blacklist; to allow only a few desktop clients to request work, use a whitelist.
  - ◆ File caching control—you can specify whether a file can be cached on the desktop client.
  - ◆ Application authorization—you can configure LSF desktop support to run only authorized applications.
  - ◆ HTTP digest authentication—to ensure secure communication between the Web Server and each desktop client, you can enable digest authentication.
- Usability**
- ◆ Tomcat and Apache are combined—only the default port (port 80) is used for both file transfer and job control.
  - ◆ New Logon and Idle desktop client modes—in Logon mode, jobs can run only when no user is logged on to the desktop client; in Idle mode, jobs can run only when the desktop client has been idle for a specified, configurable, period.
  - ◆ Enhanced Screen saver desktop client mode—desktop client users can now use their own screen saver instead of the default screen saver.

# Upgrade and Compatibility Notes

## UPGRADE document

To upgrade to LSF Version 6.1, follow the steps in [upgrade.html](#).

## API compatibility

Full backward compatibility: your applications will run under LSF Version 6.1 without changing any code.

The Platform LSF Version 6.1 API is fully compatible with the LSF Version 6.0. and 5.x APIs. An application linked with the LSF Version 6.0 or 5.x libraries will run under LSF Version 6.1 without relinking.

To take full advantage of new Platform LSF Version 6.1 features, you should recompile your existing LSF applications with LSF Version 6.1.

## Server host compatibility Platform LSF

To use new features introduced in Platform LSF Version 6.1, you must upgrade all hosts in your cluster to 6.1.

LSF 6.0 and 5.x servers are compatible with Version 6.1 master hosts. All LSF 6.0 and 5.x features are supported by 6.1 master hosts.

## Cross-product compatibility

LSF Reports supports LSF versions 5.1 and higher and License Scheduler versions 6.0 and higher.

See “[Platform LSF Family Compatibility](#)” at [www.platform.com/services/support/services/compatibility.asp](http://www.platform.com/services/support/services/compatibility.asp) for more information.

## Changed behavior

- ◆ In closer compliance with POSIX and other standards, `bjobs` no longer returns a non-zero exit code when no unfinished jobs are found. `bjobs` now returns zero (0) for all job states.
- ◆ If `LSF_MASTER_LIST` is specified in `lsf.conf`, and you run `lsadmin reconfig`, you are now prompted to confirm the reconfiguration for only the master candidate hosts:
 

```
# lsadmin reconfig

Checking configuration files ...
No errors found.

Restart only the master candidate hosts? [y/n] y
```
- ◆ `blstat -Lp lp_name` displays all license usage in the license project
 

The `-P` option for this command is no longer available and is replaced by the `-Lp` option.
- ◆ `blinfo -Lp` displays all project names that LSF License Scheduler is managing

The `-P` option for this command is no longer available and is replaced by the `-Lp` option.

- ◆ To enable dynamic host configuration, you must now define the following parameters:
  - ❖ `LSF_MASTER_LIST` and `LSF_DYNAMIC_HOST_WAIT_TIME` in `lsf.conf`
  - ❖ `LSF_HOST_ADDR_RANGE` in `lsf.cluster.cluster_name`
- ◆ If idle job exception handling is configured in your queues, `mbatchd` now checks if the job is idle after 20 minutes of run time. Jobs with idle factor is zero are no longer immediately reported as idle. Because some jobs need time to load data or do not reach full CPU utilization until after some run time, the delay avoids having jobs flagged as idle when they are running correctly.  
Set `DETECT_IDLE_JOB_AFTER` in `lsb.params` to change the 20 minute default.
- ◆ `bjobs` displays pending jobs that have been automatically requeued at the bottom of the pending job list.
- ◆ In `lsf.conf`: `LSF_DYNAMIC_HOST_TIMEOUT`

#### Obsolete parameters

## New configuration parameters and environment variables

The following new parameters and environment variables have been added for LSF Version 6.1:

**hosts** The default host file syntax

*ip\_address official\_name [alias [alias ...]]*

is powerful and flexible, but it is difficult to configure in systems where a single host name has many aliases, and in multihomed host environments.

In these cases, the `hosts` file can become very large and unmanageable, and configuration is prone to error.

The syntax of the LSF `hosts` file supports host name ranges as aliases for an IP address. This simplifies the host name alias specification.

To use host name ranges as aliases, the host names must consist of a fixed node group name prefix and node indices, specified in a form like:

*host\_name[index\_x-index\_y, index\_m, index\_a-index\_b]*

For example:

*atlasD0[0-3,4,5-6, ...]*

is equivalent to:

*atlasD0[0-6, ...]*

The node list does not need to be a continuous range (some nodes can be configured out). Node indices can be numbers or letters (both upper case and lower case).

For example, some systems map internal compute nodes to single LSF host names. A host file might contains 64 lines, each specifying an LSF host name and 32 node names that correspond to each LSF host:

```
...
177.16.1.1 atlasD0 atlas0 atlas1 atlas2 atlas3 atlas4 ... atlas31
177.16.1.2 atlasD1 atlas32 atlas33 atlas34 atlas35 atlas36 ... atlas63
...
```

In the new format, you still map the nodes to the LSF hosts, so the number of lines remains the same, but the format is simplified because you only have to specify ranges for the nodes, not each node individually as an alias:

```
...
177.16.1.1 atlasD0 atlas[0-31]
177.16.1.2 atlasD1 atlas[32-63]
...
```

#### lsb.params ◆ CONDENSE\_PENDING\_REASONS = Y | N

If enabled, condenses all host-based pending reasons into one generic pending reason.

If enabled, you can request a full pending reason list by running the following command:

```
%badmin diagnose <jobId>
```

You must be LSF administrator or a queue administrator to run this command.

For example:

❖ CONDENSE\_PENDING\_REASONS = Y

If a job has no other pending reason, `bjobs -p` or `bjobs -l` displays the following:

```
Individual host based reasons
```

❖ CONDENSE\_PENDING\_REASONS = N

The pending reasons are not suppressed. Host-based pending reasons are displayed.

#### ◆ JOB\_POSITION\_CONTROL\_BY\_ADMIN = Y | N

Allows an LSF administrator to control whether users can use `btop` and `bbot` to move jobs to the top and bottom of queues. When `JOB_POSITION_CONTROL_BY_ADMIN = Y`, only LSF administrators (including any queue administrators) can use `bbot` and `btop` to move jobs within a queue.

A user attempting to use `bbot` or `btop` receives the error message  
User permission denied.

#### ◆ DETECT\_IDLE\_JOB\_AFTER = *time\_minutes*

The minimum job run time before `mbatchd` reports that the job is idle.

By default, `mbatchd` checks if the job is idle after 20 minutes of run time.

#### ◆ MAX\_CONCURRENT\_JOB\_QUERY = *integer*

Defines how many concurrent job queries `mbatchd` can handle.

❖ If `mbatchd` is using multithreading, a dedicated query port is defined by the parameter `LSB_QUERY_PORT` in `lsf.conf`. When `mbatchd` has a dedicated query port, the value of `MAX_CONCURRENT_JOB_QUERY` sets the maximum

number of job queries for each child `mbatchd` that is forked by `mbatchd`. This means that the total number of job queries can be more than the number specified by `MAX_CONCURRENT_JOB_QUERY`.

- ❖ If `mbatchd` is not using multithreading, the value of `MAX_CONCURRENT_JOB_QUERY` sets the maximum total number of job queries.

- ◆ `MAX_PEND_JOBS = integer`

The maximum number of pending jobs in the system. This is the hard system-wide pending job threshold. No user or user group can exceed this limit unless the job is forwarded from a remote cluster.

- ◆ `MIN_SWITCH_PERIOD = seconds`

The minimum period in seconds between event log switches.

Works together with `MAX_JOB_NUM` to control how frequently `mbatchd` switches the file. `mbatchd` checks if `MAX_JOB_NUM` has been reached every `MIN_SWITCH_PERIOD` seconds. If `mbatchd` finds that `MAX_JOB_NUM` has been reached, it switches the events file.

- ◆ `PARALLEL_SCHED_BY_SLOT = y | Y`

LSF schedules jobs based on the number of slots assigned to the hosts instead of the number of CPUs. These slots can be defined by host in `lsb.hosts` or by slot limit in `lsb.resources`.

All slot-related messages still show the word `processors`, but actually refer to `slots` instead. Similarly, all scheduling activities also use slots instead of processors.

- ◆ `SUB_TRY_INTERVAL = integer`

The number of seconds for the requesting client to wait before resubmitting a pending job. This is sent by `mbatchd` to the client. The default is 60.

## lsb.queues

- ◆ New syntax for `RES_REQ`—use the OR operator (`|`) in resource requirement strings to specify multiple resource reservations and evaluate reservations in the specified order until one is satisfied. This is useful when a job can run with more than one `rusage` string and you want LSF to select the most appropriate resource reservation to meet the requirements for the job.

For example, you are running an application version 1.5 as a resource called `app_lic_v15` and the same application version 2.0.1 as a resource called `app_lic_v201`. The license key for version 2.0.1 is backward compatible with version 1.5, but the license key for version 1.5 will not work with 2.0.1.

Your jobs can run using either version of the application, but you prefer to reserve version 2.0.1 of the application. If it is not available, you can use version 1.5. To configure your queue for jobs that will try `app_lic_v201` before trying `app_lic_v15`:

```
Begin Queue
```

```
...
```

```
RES_REQ = rusage[app_lic_v201=1|app_lic_v15=1]
```

```
...
```

```
End Queue
```

Job-level resource requirement specifications (`bsub -R`) that use the `||` operator take precedence over any queue-level resource requirement specifications.

- lsf.cluster** ♦ Assign exclusive resources defined in `lsf.shared` to specific hosts in `lsf.cluster.cluster_name`. Specify an exclusive resource by prefixing the resource with an exclamation mark (!). For example, the resource `bigmem` is defined in `lsf.shared`, and is assigned as an exclusive resource for `hostE`:

```
Begin Host
...
HOSTNAME  model  type  server  rlm  mem  pg  RESOURCES
hostE      !      !      1        ()   ()   ()  (!bigmem)
...
End Host
```

A job will only be dispatched to `hostE` if `bigmem` is explicitly requested in either the job or queue level resource requirements.

- ♦ **LSF\_HOST\_ADDR\_RANGE**

The default is now `LSF_HOST_ADDR_RANGE=*. *.*.*`

No security is enabled. Any host in any domain can join the cluster dynamically if you enabled dynamic host configuration.

To enable dynamic host configuration, you must define the following parameters:

- ❖ `LSF_HOST_ADDR_RANGE` in `lsf.cluster.cluster_name`
- ❖ `LSF_MASTER_LIST` and `LSF_DYNAMIC_HOST_WAIT_TIME` in `lsf.conf`

- lsf.conf** ♦ `LSB_MAX_JOB_DISPATCH_PER_SESSION = integer`

Defines the maximum number of jobs that `mbatchd` can dispatch during one job scheduling session.

Both `mbatchd` and `sbatchd` must be restarted when you change the value of this parameter.

If set to a value greater than 300, the file descriptor limit is increased on operating systems that support a file descriptor limit greater than 1024.

Use together with `MAX_SBD_CONN` in `lsb.params`. Set `MAX_SBD_CONN` to the same value as `LSB_MAX_JOB_DISPATCH_PER_SESSION`.

For example:

- ❖ `LSB_MAX_JOB_DISPATCH_PER_SESSION = 300`  
The file descriptor limit is 1024.
- ❖ `LSB_MAX_JOB_DISPATCH_PER_SESSION = 1000`  
The file descriptor limit is greater than 1024 on operating systems that support a greater limit.

- ♦ `LSF_DYNAMIC_HOST_WAIT_TIME = time_seconds`

Defines the period of time from startup for dynamic slave LIMs (hosts) to wait for an acknowledgement from the master LIM. This signals to the dynamic host that it is already in the cluster and therefore does not need to be added. If it does not receive this acknowledgement, the dynamic host sends a request to the master LIM to add it to the cluster.

Static hosts should receive the acknowledgement from the master before this period of time, while dynamic hosts need to send a request to the master LIM to add them to the cluster.

To enable dynamic host configuration, you must define the following parameters:

- ❖ LSF\_MASTER\_LIST and LSF\_DYNAMIC\_HOST\_WAIT\_TIME in `lsf.conf`
- ❖ LSF\_HOST\_ADDR\_RANGE in `lsf.cluster.cluster_name`

The recommended value for LSF\_DYNAMIC\_HOST\_WAIT\_TIME is up to 60 seconds for every 1000 hosts in the cluster, for a maximum of 15 minutes. Selecting a smaller value will result in a quicker response time for new hosts at the expense of an increased load on the master LIM.

For example:

```
LSF_DYNAMIC_HOST_WAIT_TIME=60
```

Hosts will wait 60 seconds from startup to receive an acknowledgement from the master LIM. If it does not receive the acknowledgement within the 60 seconds, it will send a request for the master LIM to add it to the cluster.

- ◆ LSF\_LIC\_SCHED\_HOSTS = *candidate\_host\_list*

*candidate\_host\_list* is a space-separated list of hosts that are candidate License Scheduler hosts.

The candidate License Scheduler host list is read by LIM on each host to check if the host is a candidate License Scheduler master host. If the host is on the list, LIM starts up the License Scheduler daemon (bld) on the host.

- ◆ LSF\_LIC\_SCHED\_PREEMPT\_REQUEUE = Y | N

Set this parameter to requeue a job whose license is preempted by License Scheduler. The job will be killed and requeued instead of suspended.

If you set LSF\_LIC\_SCHED\_PREEMPT\_REQUEUE, do not set LSF\_LIC\_SCHED\_PREEMPT\_SLOT\_RELEASE. If both these parameters are set, LSF\_LIC\_SCHED\_PREEMPT\_SLOT\_RELEASE is ignored.

- ◆ LSF\_LIC\_SCHED\_PREEMPT\_SLOT\_RELEASE = Y | N

Set this parameter to release the slot of a job that is suspended when the its license is preempted by License Scheduler.

If you set LSF\_LIC\_SCHED\_PREEMPT\_SLOT\_RELEASE, do not set LSF\_LIC\_SCHED\_PREEMPT\_REQUEUE. If both these parameters are set, LSF\_LIC\_SCHED\_PREEMPT\_SLOT\_RELEASE is ignored.

- ◆ LSF\_LIC\_SCHED\_PREEMPT\_STOP = Y | N

Set this parameter to use job controls to stop a job that is preempted. When this parameter is set, a UNIX SIGSTOP signal is sent to suspend a job instead of a UNIX SIGTSTP.

To send a SIGSTOP signal instead of SIGTSTP, the following parameter in `lsb.queues` must also be set:

```
JOB_CONTROLS=SUSPEND[SIGSTOP]
```

- ◆ LSF\_NON\_PRIVILEGED\_PORTS=y | Y

Disables privileged ports usage.



By default, LSF daemons and clients running under root account will use privileged ports to communicate with each other. Without LSF\_NON\_PRIVILEGED\_PORTS defined, and if LSF\_AUTH is not defined in `lsf.conf`, LSF daemons check privileged port of request message to do authentication.

If LSF\_NON\_PRIVILEGED\_PORTS=Y is defined, LSF clients (LSF commands and daemons) will not use privileged ports to communicate with daemons and LSF daemons will not check privileged ports of incoming requests to do authentication.

#### Environment variables

- ◆ LSB\_EXEC\_USAGE = "*resource\_name1 resource\_value1 resource\_name2 resource\_value2...*"

Indicates which rusage string is satisfied to permit the job to run. This environment variable is necessary because the OR ( | ) operator specifies alternate rusage strings for running jobs.

- ◆ LSB\_NTRIES=integer

The number of times that LSF libraries attempt to contact `mbatchd`.

For example, if this parameter is undefined when you type `bjobs`, LSF displays `batch system not responding` if `mbatchd` cannot be contacted or if the number of pending jobs exceeds `MAX_PEND_JOBS` specified in `lsb.params`.

If this parameter is set to a value, LSF only attempts to contact `mbatchd` the defined number of times and then quits. LSF will wait for a period of time equal to `SUB_TRY_INTERVAL` specified in `lsb.params` before attempting to contact `mbatchd` again.

## New command options and output

The following command options and output have changed for LSF Version 6.1:

#### New -Lp option for License Scheduler projects

- ◆ This option is used for License Scheduler projects, while the current -P option is now used exclusively for LSF projects. This option is useful for distinguishing between LSF projects and License Scheduler projects. The -Lp option is available for the following commands:

- ❖ `bsub -Lp lp_name` explicitly assigns projects to the License Scheduler
- ❖ `bmod -Lp lp_name jobID` attaches the job to the license project
- ❖ `bmod -Lpn lp_name jobID` detaches the job from the license project
- ❖ `bjobs -Lp lp_name` displays all jobs belonging to the license project
- ❖ `bhist -Lp lp_name` displays all jobs belonging to the license project
- ❖ `bacct -Lp lp_name` displays all the jobs that ran in the license project
- ❖ `blstat -Lp lp_name` displays all license usage in the license project

The -P option for this command is no longer available and is replaced by the -Lp option.

- ❖ `blinfo -Lp` displays all project names that LSF License Scheduler is managing

The -P option for this command is no longer available and is replaced by the -Lp option.

- bbot and btop** ♦ To prevent users from changing the queue position of a pending job with bbot and btop, configure `JOB_POSITION_CONTROL_BY_ADMIN=Y` in `lsb.params`.

- badmin** ♦ `diagnose [job_ID ... | "job_ID[index]" ...]`  
 Displays full pending reason list if CONDENSE\_PENDING\_REASONS=Y is set in `lsb.params`. For example:  

```
% badmin diagnose 1057
```
- ♦ `hghostadd [-C comment] host_group host_name [host_name ...]`  
 Dynamically adds hosts to a host group. After receiving the host information from the master LIM, `mbatchd` dynamically adds the host without triggering a reconfiguration.  
 Once the host is added to the group, it will be considered to be part of that group with respect to scheduling decision making for both newly submitted jobs and for existing pending jobs.  
 This command fails if any of the specified host groups or host names are not valid.  
 To enable dynamic host configuration, define `LSF_MASTER_LIST` and `LSF_DYNAMIC_HOST_WAIT_TIME` in `lsf.conf` and `LSF_HOST_ADDR_RANGE` in `lsf.cluster.cluster_name`.
- ❖ `-C comment`  
 Logs the text of comment as an administrator comment record to `lsb.events`.  
 The maximum length of the comment string is 512 characters.
- ♦ `hghostdel [-f] [-C comment] host_group host_name [host_name ...]`  
 Dynamically deletes hosts from a host group by triggering an `mbatchd` reconfiguration. The host must be dynamic, otherwise it will not be deleted from a host group that is defined in the `lsb.hosts` file. This command fails if any of the specified host groups or host names are not valid.  
 To enable dynamic host configuration, you must define the following parameters:
- ❖ `LSF_MASTER_LIST` and `LSF_DYNAMIC_HOST_WAIT_TIME` in `lsf.conf`
- ❖ `LSF_HOST_ADDR_RANGE` in `lsf.cluster.cluster_name`
- When a dynamic host is configured as a static host in `lsf.cluster.cluster_name` run `hghostdel` to remove the host from the host group as a dynamic member.
- ❖ `-f`  
 Disables interaction and does not prompt for confirmation before forcing an `mbatchd` reconfiguration.
- ❖ `-C comment`  
 Logs the text of comment as an administrator comment record to `lsb.events`.  
 The maximum length of the comment string is 512 characters.
- bhist** ♦ If you submitted a job using the OR (`|`) expression to specify alternate resources, `-l` displays the successful usage string that caused the job to run

- bmgroup** ♦ -l displays static and dynamic host group members.  
A plus sign (+) before the host name indicates that the host is dynamic and is currently a member of the host group. A minus sign (-) before the host name indicates that the host is currently not an LSF host but is a member of the dynamic group.
- bsub** ♦ -eo *err\_file* option—overwrites the standard error output of the job to the specified file path.
- ♦ If PARALLEL\_SCHED\_BY\_SLOT=Y in `lsb.params`, the -n option specifies the number of slots required to run the job, not the number of processors.
- ♦ -oo *out\_file* option—overwrites the standard output of the job to the specified file path if it exists, or sends the output to a new file if it does not exist. Sends the output by mail if the system has trouble writing to the file.
- ♦ -R option:
- ❖ Exclusive resources need to be explicitly specified within the resource requirement string. For example, you defined a resource called `bigmem` in `lsf.shared` and defined it as an exclusive resource for `hostE` in `lsf.cluster.cluster_name`. Use the following command to submit a job that will run on `hostE`:  

```
% bsub -R "bigmem" myjob
```

or  

```
% bsub -R "defined(bigmem)" myjob
```

If the -m option is specified with a single host name, the -R option is ignored.
  - ❖ Use the OR operator (|) in resource requirement strings to specify multiple resource reservations and evaluate reservations in the specified order until one is satisfied. This is useful when a job can run with more than one `rusage` string and you want LSF to select the most appropriate resource reservation to meet the requirements for the job.  
For example, you are running an application version 1.5 as a resource called `app_lic_v15` and the same application version 2.0.1 as a resource called `app_lic_v201`. The license key for version 2.0.1 is backward compatible with version 1.5, but the license key for version 1.5 will not work with 2.0.1.  
Job-level resource requirement specifications that use the || operator take precedence over any queue-level resource requirement specifications.  
If you can only run your job using one version of the application, submit the job without specifying an alternate resource. To submit a job that will only use `app_lic_v201`:  

```
% bsub -R "rusage[app_lic_v201=1]" myjob
```

If you can run your job using either version of the application, try to reserve version 2.0.1 of the application. If it is not available, you can use version 1.5. To submit a job that will try `app_lic_v201` before trying `app_lic_v15`:  

```
% bsub -R "rusage[app_lic_v201=1|app_lic_v15=1]" myjob
```

If different versions of an application require different system resources, you can specify other resources in your rusage strings. To submit a job that will use 20 MB of memory for `app_lic_v15` or 20 MB of memory and 50 MB of swap space for `app_lic_v201`:

```
% bsub -R "rusage [mem=20:app_lic_v15=1 | mem=20:swp=50:app_lic_v201=1] " myjob
```

### lsrun, lsgrun, and lsplace

- ◆ `-R` option—exclusive resources need to be explicitly specified within the resource requirement string. For example, you defined a resource called `bigmem` in `lsf.shared` and defined it as an exclusive resource for `hostE` in `lsf.cluster.cluster_name`. Use the following command submit a task to run on `hostE`:

```
% lsrun -R "bigmem" myjob
```

or

```
% lsrun -R "defined(bigmem)" myjob
```

If the `-m` option is specified with a single host name, the `-R` option is ignored.

## New files added to installation

The following new files have been added to the Platform LSF Version 6.1 License Scheduler installation:

- ◆ `LSF_BINDIR/bcollect`
- ◆ `LSF_BINDIR/blcollect`
- ◆ `LSF_BINDIR/blhosts`
- ◆ `LSF_BINDIR/blinfo`
- ◆ `LSF_BINDIR/blstat`
- ◆ `LSF_BINDIR/blusers`
- ◆ `LSF_BINDIR/bladmin`
- ◆ `LSB_CONFDIR/cluster_name/configdir/lsf.licensescheduler`

### Symbolic links to LSF files

If your installation uses symbolic links to other files in these directories, you must manually create links to these new files.

## New accounting and job event fields

- ◆ `lsb.acct` JOB\_FINISH: `licenseProject (%s)`—LSF License Scheduler project name

## Known Issues

### Platform LSF Version 6.1

- ◆ `bjobs -A` shows the last submitted job array first. A fix is planned for this issue.
- ◆ In LSF License Scheduler, `blstat` reports incorrect token usage after `bladmin reconfig` or `badmin reconfig`. An `mbatchd` patch is needed to resolve this problem.

### Documentation changes

In LSF Version 6.1 *Administering Platform LSF* and *Platform LSF Reference* dated December 2004 describes The `LSF_HOST_NET_FILTER` parameter in `lsf.conf`. This parameter is not currently supported. Documentation dated April 2005 corrects this error.

# Learning About Platform LSF Version 6.1

## Finding Platform LSF information

Information about Platform LSF is available online from the following sources:

- ◆ “World Wide Web and FTP”
- ◆ “LSF README, release notes, and upgrade”
- ◆ “Platform LSF Family documentation”

## World Wide Web and FTP

The latest information about all supported releases of Platform LSF is available on the Platform Web site at [www.platform.com](http://www.platform.com).

If you have problems accessing the Platform web site or the Platform FTP site, send email to [support@platform.com](mailto:support@platform.com).

Visit the Platform User Forum at [www.platformusers.net](http://www.platformusers.net) to discuss distributed workload management and Grid Computing.

## LSF README, release notes, and upgrade

Before downloading and installing LSF, be sure to read the files named [readme.html](#) and [release\\_notes.html](#).

To upgrade to LSF Version 6.1, follow the steps in [upgrade.html](#).

## Platform LSF Family documentation

The FTP documentation directory [/distrib/6.1/docs/](#) contains the LSF documentation set in HTML and PDF format.

### Download or view LSF documentation online

View or download the LSF documentation in HTML or PDF format from:

- ◆ Platform Computing Web site:  
[www.platform.com/lsf\\_docs](http://www.platform.com/lsf_docs)
- ◆ Platform Computing FTP site ([ftp.platform.com](ftp://ftp.platform.com)):  
[/distrib/6.1/docs/](#)

### Platform LSF

Title	PDF	HTML .zip	.tar.Z
<b>Installing and upgrading</b>			
<a href="#">README for Platform LSF Version 6.1</a>	395 KB		
<a href="#">Release Notes for Platform LSF Version 6.1</a>	395 KB		
<a href="#">Installing Platform LSF on UNIX and Linux (lsfinstall)</a>	195 KB	40 KB	60 KB
<a href="#">Upgrading Platform LSF on UNIX and Linux</a>	250 KB		
<b>Using</b>			
<a href="#">Running Jobs with Platform LSF</a>	775 KB	140 KB	95 KB
<a href="#">Platform LSF Desktop Support User's Guide</a>	280 KB	—	—

Title	PDF	HTML	
		.zip	.tar.Z
Administering			
Platform LSF Administrator’s Primer	750 KB	125 KB	175 KB
Administering Platform LSF	5435 KB	795 KB	1073 KB
Platform LSF Reference	4790 KB	560 KB	700 KB
Platform LSF Quick Reference	125 KB	—	—
Using Platform LSF on Windows	1310 KB	200 KB	255 KB
Licensing Platform LSF	640 KB	90 KB	135 KB
Platform LSF Cluster Management Tools	445 KB	50 KB	70 KB
Platform LSF Desktop Support Administrator’s Guide	1510 KB	—	—
Extending			
Using Platform LSF MultiCluster	930 KB	80 KB	110 KB
Using Platform LSF License Scheduler	885 KB	210 KB	290 KB
Using the Platform LSF SDK	1285 KB	210 KB	275 KB
Platform LSF API Reference	1755 KB	620 KB	290 KB
Using Platform LSF Make	175 KB	—	—
Complete Platform LSF Version 6.1 HTML Doc Set		3075 KB	3580 KB
Complete Platform LSF Version 6.1 PDF Doc Set		11415 KB	13895 KB

## LSF Integrations

<a href="#">Integrating Platform LSF and Avaki Data Grid</a>	320 KB
<a href="#">Generic MPI Framework for Platform LSF HPC</a>	180 KB
<a href="#">Using Platform LSF with Rational ClearCase</a>	335 KB
<a href="#">Using Platform LSF Frame Arrays</a>	325 KB
<a href="#">Using Platform LSF with SNMP</a>	170 KB

## Platform LSF Reports

The LSF Reports documentation set includes the following:

- ◆ *Platform LSF Reports Installation Guide* describes how to install LSF Reports and the Firebird database.
- ◆ *Platform LSF Reports Administrator's Guide* describes how to use the LSF Reports console to perform administrative tasks, and change the default data collection, data processing, and display settings.
- ◆ *Platform LSF Reports User's Guide* describes the reports viewed by end-users.
- ◆ *Platform LSF Reports Readme* contains last-minute updates and known issues.
- ◆ *Platform LSF Reports Online Help* provides context-sensitive help for the LSF Reports console.



## Platform LSF Analytics

The LSF Analytics documentation set includes the following:

- ◆ *Platform LSF Analytics Installation Guide—UNIX* describes how to install the LSF Analytics on a UNIX host.
- ◆ *Platform LSF Analytics Installation Guide—Windows* describes how to install LSF Analytics on a Windows host.
- ◆ *Platform LSF Analytics Administrator's Guide* describes how to use the LSF Analytics console to perform administrative tasks, and change the default data collection, data processing, and display settings.
- ◆ *Platform LSF Analytics User's Guide* describes the cubes and reports viewed by end-users.
- ◆ *Platform LSF Analytics Readme* contains last-minute updates and known issues.
- ◆ *Platform LSF Analytics Online Help* provides context-sensitive help for the LSF Analytics console.

## Platform training

Platform's Professional Services training courses can help you gain the skills necessary to effectively install, configure and manage your Platform products. Courses are available for both new and experienced users and administrators at our corporate headquarters and Platform locations worldwide.

Customized on-site course delivery is also available.

Find out more about [Platform Training](#) at [www.platform.com/training](http://www.platform.com/training), or contact [Training@platform.com](mailto:Training@platform.com) for details.

# Getting Technical Support

## Contacting Platform

Contact Platform Computing or your LSF vendor for technical support. Use one of the following to contact Platform technical support:

**Email** [support@platform.com](mailto:support@platform.com)

**World Wide Web** [www.platform.com](http://www.platform.com)

**Mail** Platform Support  
Platform Computing Corporation  
3760 14th Avenue  
Markham, Ontario  
Canada L3R 3T7

When contacting Platform, please include the full name of your company.

See the Platform Web site at [www.platform.com/contactus](http://www.platform.com/contactus) for other contact information.

## Patch updates and other notifications

To get periodic patch update information, critical bug notification, and general support notification from Platform Support, contact

[supportnotice-request@platform.com](mailto:supportnotice-request@platform.com) with the subject line containing the word "subscribe".

To get security related issue notification from Platform Support, contact [securenotice-request@platform.com](mailto:securenotice-request@platform.com) with the subject line containing the word "subscribe".

## We'd like to hear from you

If you find an error in any Platform documentation, or you have a suggestion for improving it, please let us know:

**Email** [doc@platform.com](mailto:doc@platform.com)

**Mail** Information Development  
Platform Computing Corporation  
3760 14th Avenue  
Markham, Ontario  
Canada L3R 3T7

Be sure to tell us:

- ◆ The title of the manual you are commenting on
- ◆ The version of the product you are using
- ◆ The format of the manual (HTML or PDF)

# Copyright

© 1994-2005 Platform Computing Corporation

All rights reserved.

Although the information in this document has been carefully reviewed, Platform Computing Corporation (“Platform”) does not warrant it to be free of errors or omissions. Platform reserves the right to make corrections, updates, revisions or changes to the information in this document.

UNLESS OTHERWISE EXPRESSLY STATED BY PLATFORM, THE PROGRAM DESCRIBED IN THIS DOCUMENT IS PROVIDED “AS IS” AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL PLATFORM COMPUTING BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION ANY LOST PROFITS, DATA, OR SAVINGS, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS PROGRAM.

## Document redistribution policy

This document is protected by copyright and you may not redistribute or translate it into another language, in part or in whole.

## Internal redistribution

You may only redistribute this document internally within your organization (for example, on an intranet) provided that you continue to check the Platform Web site for updates and update your version of the documentation. You may not make it available to your organization over the Internet.

## Trademarks

® LSF is a registered trademark of Platform Computing Corporation in the United States and in other jurisdictions.

™ ACCELERATING INTELLIGENCE, THE BOTTOM LINE IN DISTRIBUTED COMPUTING, PLATFORM COMPUTING, and the PLATFORM and LSF logos are trademarks of Platform Computing Corporation in the United States and in other jurisdictions.

UNIX is a registered trademark of The Open Group in the United States and in other jurisdictions.

Other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

